AMENDMENTS TO THE CLAIMS

Please amend the claims as follows:

1. (Cancelled)

2. (Previously Presented) A security hole diagnostic system comprising:

a script accumulation unit accumulating a plurality of scripts in a programming language describing procedures usually used by attackers for illegal access;

an operation unit making a request for a list of the plurality of scripts upon entry from a user;

a script control unit retrieving each script from the script accumulation unit upon the request from the operation unit, creating a list of input/output parameters, a script execution condition and a test procedure described thereby, and presenting the list of the plurality of scripts to the user, and executing a script that is selected by the user;

a plugin accumulation unit accumulating plugins with logics for attacking individual security holes;

a plugin control unit, which is called by an execution of the script by the script control unit, for retrieving from the plugin accumulation unit a plugin that is specified by the script to be executed and executing the plugin on a test target computer;

a springboard simulation program including a communications relay function, a packet transmission/reception function, a process start/end function, a function to input/output data to/from a process, and a file transfer function; and

a springboard simulation program control unit executing the plugin on the test target computer via the springboard simulation program upon instruction from the plugin.

3. (Previously Presented) The security hole diagnostic system according to claim 2, wherein the script is constructed to have a function to allow it to call another script.

4. (Previously Presented) The security hole diagnostic system according to claim 2, wherein the script includes class concept, and

wherein the script is constructed to have a function to allow it to call another script by specifying a class name when calling the another script.

5. (Cancelled)

6. (Currently Amended) A security hole diagnostic system comprising:

a script accumulation unit accumulating a plurality of scripts in a programming language describing procedures usually used by attackers for illegal access;

an operation unit making a request for a list of the plurality of scripts upon entry from a user;

a script control unit retrieving each script from the script accumulation unit upon the request from the operation unit, creating a list of input/output parameters, a script execution condition and a test procedure described thereby, and presenting the list of the plurality of scripts to the user, and executing a script that is selected by the user;

a plugin accumulation unit accumulating plugins with logics for attacking individual security holes;

a plugin control unit, which is called by an execution of the script by the script control unit, for retrieving from the plugin accumulation unit a plugin that is specified by the script to be executed and executing the plugin on a test target computer; and

a knowledge sharing unit verifying whether the script execution condition is met, wherein the knowledge sharing unit includes,

a deduction unit deriving new knowledge from information collected in an execution process of the script based on a deduction rule~~The security hole diagnostic system according to claim 5~~, and further wherein the knowledge sharing unit is constructed to have a function to execute a script for acquiring knowledge based on the deduction rule when shared knowledge is insufficient.

7. (Previously Presented) The security hole diagnostic system according to claim 2, wherein the script control unit, the plugin accumulation unit, the plugin control unit, the script accumulation unit, and the springboard simulation program control unit form a test execution unit, and the test execution unit and the operation unit are disposed separately on a network.

8. (Previously Presented) The security hole diagnostic system according to claim 2, wherein the plugin is described in an interpreter language.

9. (Previously Presented) The security hole diagnostic system according to claim 2, wherein the springboard simulation program control unit is constructed by using a protocol designed to pass firewalls.

10. (Previously Presented) The security hole diagnostic system according to claim 2, said script control unit also adding new and updated scripts to said script accumulation unit at the direction of the user.

11. (Previously Presented) The security hole diagnostic system according to claim 2, said script control unit also executing a script that is called by another script.

12. (Previously Presented) The security hole diagnostic system according to claim 2, wherein the communications relay function communicates with a second springboard simulation program.

13. (Previously Presented) The security hole diagnostic system according to claim 2, wherein the communications relay function communicates with a springboard simulation program control unit over a network.

14. (Previously Presented) The security hole diagnostic system according to claim 2, wherein the communications relay function transmits an incoming control message to the operation unit.

15. (Previously Presented) The security hole diagnostic system according to claim 2, wherein the operation unit transmits an outgoing or misdirected control message through the communications relay function.

16. (Previously Presented) The security hole diagnostic system according to claim 2, said script execution condition comprising a predicate calculus based description of the conditions required for executing the script.

17. (Currently Amended) The security hole diagnostic system according to ~~claim 5~~claim 6, wherein said knowledge sharing unit determines whether script execution conditions have been met and communicates said determination to said script control unit.

18. (Previously Presented) The security hole diagnostic system according to claim 7, wherein the test execution unit is disposed outside of a firewall, and the operation unit is disposed inside of a firewall.

19. (Previously Presented) The security hole diagnostic system according to claim 2, said plugins being editable while a diagnostic script is running.

20. (Currently Amended) A method for detecting security holes in a computer system comprising:

accumulating a plurality of scripts in a programming language describing procedures used by attackers for illegal access;

retrieving at a list of least one of said plurality of scripts from a script accumulation unit upon a user-initiated request from an operation unit;

creating a list of input/output parameters, a script execution condition and a test procedure described thereby;

presenting the list of retrieved scripts to the user, and executing a script that is selected by the user;

accumulating plugins with logics for attacking individual security holes; and

retrieving from a plugin accumulation unit at least one plugin that is specified by the script to be executed and executing the plugin on a test target computer; and

verifying whether the script execution condition is met by deriving new knowledge from information collected in an execution process of the script based on a deduction rule, where deriving new knowledge includes executing a script for acquiring knowledge based on the deduction rule when shared knowledge is insufficient.

21. (Currently Amended) A computer readable medium having embodied thereon a program for detecting security holes in a computer system which, when executed, performs the steps of:

accumulating a plurality of scripts in a programming language describing procedures used by attackers for illegal access;

retrieving at a list of least one of said plurality of scripts from a script accumulation unit upon a user-initiated request from an operation unit;

creating a list of input/output parameters, a script execution condition and a test procedure described thereby;

presenting the list of retrieved scripts to the user, and executing a script that is selected by the user;

accumulating plugins with logics for attacking individual security holes; and

retrieving from a plugin accumulation unit at least one plugin that is specified by the script to be executed and executing the plugin on a test target computer; and

verifying whether the script execution condition is met by deriving new knowledge from information collected in an execution process of the script based on a deduction rule, where deriving new knowledge includes executing a script for acquiring knowledge based on the deduction rule when shared knowledge is insufficient.